

Article

LSO-FastSLAM: A New Algorithm to Improve the Accuracy of Localization and Mapping for Rescue Robots

Daixian Zhu ^{1,*}, Yanan Ma ¹, Mingbo Wang ^{2,*}, Jing Yang ³, Yichen Yin ¹ and Shulin Liu ⁴

¹ College of Communication and Information Engineering, Xi'an University of Science and Technology, Xi'an 710054, China; crazyboymyn@163.com (Y.M.); jetyyc@163.com (Y.Y.)

² College of Energy Science and Engineering, Xi'an University of Science and Technology, Xi'an 710054, China

³ Xi'an Xiangteng Microelectronics Technology Co., Ltd., Xi'an 710054, China; crazygirlyj@163.com

⁴ College of Electrical and Control Engineering, Xi'an University of Science and Technology, Xi'an 710054, China; lsigma@163.com

* Correspondence: zhudaixian@xust.edu.cn (D.Z.); Lqs_wmb@163.com (M.W.)

Abstract: This paper improves the accuracy of a mine robot's positioning and mapping for rapid rescue. Specifically, we improved the FastSLAM algorithm inspired by the lion swarm optimization method. Through the division of labor between different individuals in the lion swarm optimization algorithm, the optimized particle set distribution after importance sampling in the FastSLAM algorithm is realized. The particles are distributed in a high likelihood area, thereby solving the problem of particle weight degradation. Meanwhile, the diversity of particles is increased since the foraging methods between individuals in the lion swarm algorithm are different so that improving the accuracy of the robot's positioning and mapping. The experimental results confirmed the improvement of the algorithm and the accuracy of the robot.

Keywords: mine robot; simultaneous localization and mapping; FastSLAM; lion swarm optimization algorithm; particle weight; particle diversity



Citation: Zhu, D.; Ma, Y.; Wang, M.; Yang, J.; Yin, Y.; Liu, S.

LSO-FastSLAM: A New Algorithm to Improve the Accuracy of Localization and Mapping for Rescue Robots. *Sensors* **2022**, *22*, 1297. <https://doi.org/10.3390/s22031297>

Academic Editor: Hyun Myung

Received: 3 December 2021

Accepted: 3 February 2022

Published: 8 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Mine rescue robot plays a critical role in mine rescues [1–3]. When a mine accident occurs, it can replace the rescue workers to search and rescue so as to protect rescuers from the great amount of combustible gas and dust in the mine. In particular, due to the complex environment after the disaster, rescuers cannot enter the narrow space, so they need to use robots to search and rescue. Moreover, the robot can determine its own position through the map established by itself and guide the trapped people to leave the site after the disaster. Hence, an accurate map under the mine has great significance for fast rescue. However, when the mine accident occurs, the underground environment of the mine changes greatly, and the original underground map is not suitable for the current rescue environment. Although the related positioning system is equipped under the mine, the positioning equipment is damaged, and the precise positioning function cannot be realized when the mine accident occurs. Therefore, simultaneous localization and mapping (SLAM) of robots play a significant role in mine rescue.

The concept of SLAM is proposed to deal with the problem of simultaneous positioning and mapping of robots [4–6]. In an unknown environment, the robot combines its own sensors such as lidar and vision cameras to estimate its own pose and build an environment map [7,8]. The classical SLAM algorithm research is based on filtering theory, which includes Kalman filter-based EKF-SLAM [9] and particle filter-based FastSLAM algorithm [10].

Among them, EKF-SLAM can only handle Gaussian noise, and its processing speed is slow for non-Gaussian noise environment. Moreover, in the same scenario, its processing speed is slower than the FastSLAM algorithm.

However, there are some potential defects in the FastSLAM algorithm. One of the problems is particle weight degradation and particle diversity loss. This problem is caused by the adoption of Rao-Blackwellized filter in the FastSLAM algorithm, which will decrease the accuracy of the FastSLAM algorithm [11–15]. Another problem of the FastSLAM algorithm is that it requires a large number of particles to maintain the accuracy of the algorithm in a complex environment, which will increase the running time of the algorithm and reduce the efficiency of the robot. In addition, the accuracy of algorithm decreased once the number of particles is insufficient [16].

In order to overcome the drawbacks of the FastSLAM algorithm, researchers put forward different ideas for algorithm improvement. Kim, together with his coworkers, proposed the unscented FastSLAM (UFastSLAM) algorithm. Based on the FastSLAM algorithm, they used an unscented Kalman filter (UKF) instead of an extended Kalman filter (EKF) for robot pose estimation, which improved the filtering accuracy of the robot [17]. However, they did not solve the particle degradation problem. HeBo et al. solved the problem of particle weight degradation in the UFastSLAM algorithm through particle swarm optimization (PSO) algorithm, and tested it on their autonomous underwater vehicle (AUV), which verified the effectiveness and feasibility of the improved algorithm [18]. Different from the above methods, with the deepening of the research on the meta-heuristic algorithm [19–23], more and more researchers use the meta-heuristic algorithm to improve the FastSLAM algorithm [24–28].

Tian and his coworkers optimized the FastSLAM through the improved firefly algorithm to improve the robot positioning with the accuracy of mapping. Meanwhile, this method effectively solved the problem of particle degradation [29]. Tian et al. used the meta-priming algorithm to optimise the importance sampling of the FastSLAM algorithm, thus effectively solving the degradation of particle weights and loss of particle diversity, and improved the robot localization mapping accuracy. In contrast, some literature [25] focused on the optimization of the resampling process in the FastSLAM algorithm, and optimized the resampling process of the FastSLAM algorithm by the particle swarm algorithm to achieve the increased localization mapping accuracy of the robot. Although the metaheuristic algorithm has achieved impressive results in optimising the FastSLAM algorithm, the algorithm is prone to fall into local optimality due to the strong optimisation capability of the metaheuristic algorithm; in addition, using the metaheuristic algorithm to optimise the FastSLAM algorithm, the fusion of multiple algorithms will lead to a significant increase in the algorithm time complexity.

In order to solve the problem of low accuracy of mine rescue robot positioning and mapping, this paper uses the FastSLAM2.0 algorithm to decompose the robot positioning and mapping into a moving part and a conditional map part to reduce the sampling space. However, due to the degradation of particle weights and the loss of particle diversity in the FastSLAM2.0 algorithm, the accuracy of robot positioning and mapping is reduced. Based on reading a large number of the FastSLAM algorithm improvement documents, this article adopts a new meta-heuristic algorithm lion swarm optimization (LSO) algorithm [30], to optimize the FastSLAM2.0 algorithm. In our work, we optimized the particle distribution after importance sampling through the LSO algorithm and regarded the sample particle set as a lion group, and optimized the individuals' movement law between the populations in the lion group algorithm. The sample particle distribution solves particle degradation. At the same time, in order to keep particle diversity, we uses genetic algorithm to replace the lioness movement process to increase the particle diversity, and finally achieve the degradation of the particle weight of the FastSLAM2.0 algorithm. This solution to the loss of particle diversity can improve the accuracy of robot positioning and mapping.

The specific contributions of this paper are as follows:

- This paper designs and implements a new scheme to optimise the FastSLAM algorithm by means of the Lion Swarm algorithm.
- In the process of optimizing the FastSLAM algorithm through the lion swarm algorithm, the distribution of the particle set after important sampling in the FastSLAM

algorithm is achieved through the division of labour between different individuals in the lion swarm optimisation algorithm, so that the particles are distributed in the high likelihood region, and solving the particle weight degradation problem. In addition, to ensure particle diversity, a genetic algorithm is used instead of the lioness movement process in the lion swarm to further increase the particle diversity.

- In this paper, the innovative FastSLAM algorithm is applied to a rescue robot by optimizing the Lion Swarm algorithm, aiming to improve the localization and mapping accuracy of the rescue robot.

The rest of the paper is organized as follows. Section 2 introduces the FastSLAM briefly. Section 3 introduces the principle of lion swarm optimization algorithm. Section 4 discusses the proposed LSO-FastSLAM in detail. Section 5 gives the simulation results and analyzes the performance of the LSO-FastSLAM in detail. Finally, Section 6 concludes the paper and provides some suggestions for future work.

2. Background of the FastSLAM

Before introducing the FastSLAM algorithm, we will briefly describe the SLAM problem. SLAM means that the robot determines its trajectory through its own sensors in a location environment and at the same time constructs an environmental map. The robot SLAM problem can be regarded as a kind of probability problem. The robot can achieve the best estimation of pose state and environment map by observing and controlling quantity. Its expression is as follows:

$$p(x_t, m, z_t, u_t, n_t) \quad (1)$$

where x_t represents the robot's position at time t , m represents the map, z_t and u_t are the measured quantity and the controlled quantity, and n_t represents the data association. Meanwhile, the SLAM problem can be represented by a dynamic Bayesian network in Figure 1.

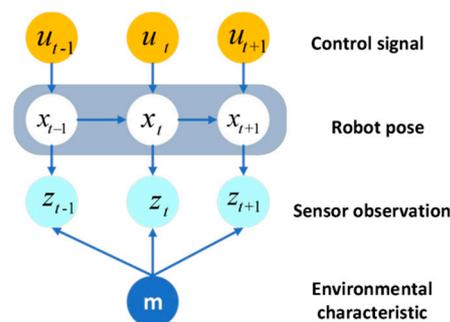


Figure 1. Dynamic Bayesian networks for SLAM.

FastSLAM is an excellent algorithm for solving SLAM problems. In FastSLAM, based on the Rao-Blackwellized particle filter, Equation (1) is decomposed into robot path estimation and independent feature estimation. Therefore, the decomposition of Equation (1) can be expressed as:

$$p(x_t, m | z_t, u_t, n_t) = p(x_t | z_t, u_t, n_t) \prod_{k=1}^M p(m_k | x_t, z_t, u_t, n_t) \quad (2)$$

where $p(x_t | z_t, u_t, n_t)$ represents the estimation of the pose of the robot, and $p(m_k | x_t, z_t, u_t, n_t)$ represents the estimation of the environmental features. In the FastSLAM algorithm, the robot pose is estimated by PF, and the environment features are estimated by the extended Kalman filter (EKF). In this process, each particle corresponds to a robot path estimation and environmental features. The particles in FastSLAM can be expressed as:

$$X_k^i = \left\{ x_k^i, u_{1,k}^i, \sum_{1,k}^i \cdots u_{N,k}^i, \sum_{N,k}^i \right\} \quad (3)$$

where i is the particle index, x_k^i is the robot path estimation, $u_{n,k}^i$ and $\Sigma_{N,k}^i$ is the mean value and covariance of the n th feature of the k th particle.

The equations above are the key mathematical point of view of the FastSLAM algorithm, for the detailed proof of the FastSLAM algorithm, according to previous literature [31]. Since the proposed distribution function selection method is different, the FastSLAM algorithm is divided into the FastSLAM1.0 algorithm and FastSLAM2.0 algorithm. The FastSLAM1.0 algorithm uses the robot motion model as the particle sampling function, so when the motion control input error is large, the estimated state of the system will be inaccurate. In contrast, in the FastSLAM2.0 algorithm, a complete EKF iterative process is first adopted, in which the latest moment of control input and landmark characteristics measurement values are integrated, and then the posterior estimates of robot pose state are used as the particle sampling function, thus improving the estimation accuracy of the robot.

Although the FastSLAM2.0 algorithm adopted a new importance density function, but it still exists particle degradation problems. Aiming to solve this problem, we introduce the resampling strategy. Although this method can solve the problem of particle weight degradation, it also can cause particle diversity loss problems. Therefore, the key issue to improve the accuracy of robot positioning and mapping is the drop in particle weight and the loss of particle diversity. This article improves on the FastSLAM2.0 algorithm.

For the derivation of the FastSLAM2.0, readers can refer to the literature [16,31]. The main steps and formulas of the FastSLAM2.0 are as follows.

- (1) Sampling the pose

$$x_t^i \sim p(x_t | x^{t-1,i}, z^t, u^t, n^t) \quad (4)$$

- (2) EKF updates the observed landmark estimates.

- (3) Importance weight calculation:

$$w_t^i = \frac{\text{target distribution}}{\text{proposal distribution}} = \frac{p(x^{t,i} | z^t, u^t, n^t)}{p(x^{t-1,i} | z^{t-1}, u^{t-1}, n^{t-1}) p(x_t^i | x^{t-1,i}, z^t, u^t, n^t)} \quad (5)$$

- (4) Re-sampling.
- (5) Unknown data associations.
- (6) Feature management.

3. Lion Swarm Optimization Algorithm

Lion Swarm Optimization Algorithm (LSO) is a new method to solve the global optimization problem of the objective function. This method completely simulates the lion's foraging behavior, migration, and population change to solve the problem [30].

In the LSO algorithm, the pride population is divided into the Lion King, the Lioness, and the Cub. The lion is responsible for guarding the territory and protecting the cubs as well as distributing food. The Lioness hunted and raised the cubs. The cubs, also known as follow lions, feed near the lion when they are hungry. After feeding, they learn to hunt from the Lioness. When he grows up, he will be driven out of his territory and becomes a stray lion. After training, the male lion in the stray lion will challenge the status of the original lion. The LSO algorithm is given in the following steps:

Initialization: Like other intelligent group algorithms, the LSO algorithms realize the initialization of the population number N , dimensional space D , individual position x_t , and the distribution probability of different individuals within the lion group.

In the literature [30], the different types of individuals in a lion group are set as follows. In a lion group, the number of adult lion individuals affects the effect of the algorithm—the greater the number of young lion individuals, the greater the difference in population numbers. The algorithm detects strong ability. Moreover, the convergence speed of the algorithm is related to the adult lion. In order to maintain the convergence of the algorithm,

the proportion of adult lions β is a random number between 0 and 0.5. Similarly, the proportion of cubs is listed as $1 - \beta$.

The lion guardians: When the LSO algorithm is used to solve the optimization problem, the location of each food source represents a feasible solution to the optimization problem, and the size of the adaptive value represents the quality of the solution. The maximum moderate value represents the Lion King. The lion retains his privileges in the best food range. Its motion formula is as follows:

$$x_i^{k+1} = g^k(1 + \gamma||P_i^k - g^k||) \quad (6)$$

where x_i^{k+1} represents the new position of the Lion King after the movement, g^k represents the optimal position in the k th generation group, and γ is a random number generated according to a normal distribution $N(0,1)$; P_i^k is the most historical record of the i th lion in the k th generation.

Lioness Hunt: In the process of the Lioness, the two Lioness cooperate to complete the hunt, then the position of the Lioness after cooperative hunting:

$$x_i^{k+1} = \frac{P_i^k + P_c^k}{2}(1 + \alpha_f \gamma) \quad (7)$$

where P_i^k is the historical optimal position of the k th lion in the i generation. P_c^k is the historical best position of a hunting partner randomly selected from the k generation lioness group and γ is the random number generated according to the normal distribution $N(0,1)$.

The α_f in the above formula represents the disturbance factor of the Lioness's moving range. The reason for setting the disturbance factor is described in detail in reference [29], and the setting formula is as follows:

$$\alpha_f = step \cdot \exp\left(-\frac{30t}{T}\right)^{10} \quad (8)$$

where $\bar{g}^k = \overline{low} + \overline{high} - g^k$ represents the maximum moving step length in the range of activity, \overline{high} and \overline{low} represents the minimum and maximum lion's moving values in the range of activity. T is the maximum number of iterations of the population, and t is the number of iterations of the current population.

Lion Cubs Follow: There are three main activities for kindergarten teachers in the lion group. (1) When they are hungry, they will turn to the Lion King; (2) When they are full, they will hunt with the Lioness; (3) When they grow up, they will be driven out of the territory by the Lion King and become A stray lion. The male lion among the lions will challenge the status of the original Lion King. Therefore, the position of the Cub after moving can be represented as:

$$x_i^{k+1} = \begin{cases} \frac{g^k + P_i^k}{2}(1 + \alpha_c \gamma), & q \leq \frac{1}{3} \\ \frac{P_m^i + P_i^k}{2}(1 + \alpha_c \gamma), & \frac{1}{3} \leq q \leq \frac{2}{3} \\ \frac{\bar{g}^k + P_i^k}{2}(1 + \alpha_c \gamma), & \frac{2}{3} \leq q \leq 1 \end{cases} \quad (9)$$

where γ, P^k, g^k represent the same meaning as Equations (4) and (5), P_m^i represents the best position, in the history of the i generation of cubs following lionesses, $\alpha_c = step\left(\frac{T-t}{T}\right)$ represents the movement disturbance factor of the Cub, and $\bar{g}^k = \overline{low} + \overline{high} - g^k$ represents the lion cub being driven away from the Lion King.

Location update: The greedy rule is used to select the individual positions of the lions before and after the update, and the global optimal is updated.

The above description is the main steps of the lion group algorithm. The lion group algorithm sets the Lion King, Lioness, cubs, and simulates the life tyle of the lion group by

appropriate values to complete the optimal solution of the problem. Its global convergence and robustness can effectively avoid the premature problem, especially for multi-peak and high-dimensional complex functions.

4. Lion Swarm Optimization Algorithm Improves FastSLAM

In the first section, particle mass degradation and particle diversity loss exist in FastSlam2.0 algorithm, where the mobile robot positioning map accuracy caused serious impact. In the literature [17–29], researchers proposed different improvement strategies. One was to apply the global optimal value algorithm to change the particle set to avoid falling into the local optimal. The common improved method is the combination of intelligent swarm algorithm and the particle filter, which attracts and repels particles through the global optimal value so that particles are concentrated and distributed in the high likelihood region, improving the filtering accuracy, particle degradation and dilution of the algorithm as results. However, this kind of algorithm has the risk of prematurity during algorithm iteration. The other is to improve the particle weight degradation and diversity loss by optimizing or replacing the particle resampling strategy, thereby improving the filtering performance of the algorithm. However, in complex environments and scenes with strong noise, a large number of particles are required to achieve state estimation, which may lead to long running time and low operating efficiency. Inspired by the work in the literature [29], this paper realized the optimization of the FastSLAM algorithm through the improved lion swarm optimization algorithm. The key idea is to optimize the particle distribution after important sampling by the FastSLAM algorithm through LSO, to make the particle distribution after important sampling into a high likelihood region, thus solving the particle weight reduction and improving the positioning accuracy of the robot.

At the same time, the survival strategies of all kinds of individuals in the lion swarm algorithm were improved and optimized. After the improvement of the Lion King position updating and the Cub following strategy, the optimization ability of particles was further enhanced. The lioness hunting process was optimized by a genetic algorithm, so as to avoid the local optimal situation of the improved algorithm. Furthermore, the degradation of particle weight and loss of particle diversity are proved to improve the positioning accuracy of the robot. The main improvement steps of the algorithm are as follows.

A. Improved Lion position update strategy

In the original LSO algorithm, the Lion King uses Equation (4) to update the position. In this process, the Lion King moves at the optimal moderate value to maintain his privileges. If the original position update formula is directly introduced into the improved algorithm, it will inevitably cause the appropriate value corresponding to the newly generated Lion King position to be lower than the original appropriate value. This will lead to a waste of computing resources and affect the positioning accuracy of the robot. This paper was inspired by literature [32] to reset the new strategy of lion position updating.

First, a set with the number $N + 1$ centered on the current position of the lion is constructed, the set settings are as follows:

$$x_k^i \sim [x_k^i, x_k^i \pm j\Delta] (j = 1, 2 \dots \frac{N}{2}, \Delta = \overline{high} \cdot 10^{-4}) \quad (10)$$

where N represents the number of particles, Δ represents the move step, and \overline{high} represents the distance of the particle with the longest distance from the global optimal value.

Secondly, set the step size threshold Δ_{max} , judge the moving step size Δ . When $\Delta \geq \Delta_{max}$, take the moving step size as $\Delta = \Delta_{max}$, instead, take the current move step as the Lion's move step. Through this step, the automatic adaptation of the step length is realized. In the early stage of the algorithm, the distance between each particle is relatively large, and the movement step length obtained by calculation must be greater than the threshold. The threshold is used as the current movement step to ensure that the Lion King is performed at a small accurate update range, and when the algorithm enters the later

stage, the particle spacing is reduced, and the moving step is smaller than the threshold, and the current moving step is used as the Lion King's moving step to ensure the Lion King's update.

Finally, the weight corresponding to each individual in the new particle set is calculated, and the individual with the largest weight is selected as the current new Lion King. Through the new Lion King update strategy, it is ensured that every time the Lion King forages, an individual better than the current Lion King will be generated, and then the particle collection will be moved to the high-likelihood area, thereby improving the filtering accuracy of the algorithm.

B. Reset the Lioness Hunting Method

In the original LSO algorithm, the lioness hunts by two lionesses hunt together, and the two lionesses are in the same position after hunting. When this step is applied to the improvement of the FastSLAM algorithm, in the late running stage of the algorithm, the distance between the particles is relatively close, so adopting this step will further reduce the diversity of particles. In this paper, the crossover step in the genetic algorithm is used as an improved lioness hunting method. The equation is as follows:

$$\tilde{x}_k^m = ax_k^m + (1 - a)x_k^n \quad (11)$$

$$\tilde{x}_k^n = ax_k^n + (1 - a)x_k^m \quad (12)$$

In Equations (11) and (12), \tilde{x}_k^m and \tilde{x}_k^n respectively represent the positions of the two lionesses participating in hunting, x_k^m and x_k^n represent the positions corresponding to the two lionesses after hunting, and a represents the crossover probability, which is 0.7.

C. Cub Follow Formula Selection

In the Cub follow Equation (7), the cub position update has three different strategies, which are moving to the Lion King, following the Lioness, and staying away from the lion group. In order to improve the filtering accuracy of the robot, in this paper, the position of the Cub is set to be updated as the Cub moves towards the Lion King, and its position updating equation is shown below.

$$x_i^{k+1} = \frac{g^k + p_i^k}{2} (1 + \alpha_c \gamma), q \leq \frac{1}{3} \quad (13)$$

The parameter setting of Equation (13) is the same as that of Equation (9). By improving following strategy, the particles represented by the young lion are concentrated and distributed in the Gauss natural region in the FastSLAM algorithm after the important sampling of particles is completed. Thus, the positioning and mapping accuracy of the robot is improved.

The improved algorithm flow chart (Figure 2) is shown below:

The above diagram shows the flow chart of the improved algorithm proposed in this paper, in which the blue part is the optimisation of the importance sampling process through the Lioness algorithm, and its detailed process is as follows: firstly, the weights of each particle are calculated, and the composition of the Lioness is designed according to the size of the weights, where the Lioness represents the globally optimal particle, the Lioness consists of particles with good fitness values, and the Cub represents the particle with poor fitness values, and then the optimization of the Lioness is achieved through the three steps above. A. Improved Lion position update strategy, B. Reset the Lioness Hunt-ing Method, C. Cub Follow Formula Selection three steps to achieve the update of the lion population and thus the optimization of the set of particles after importance sampling.

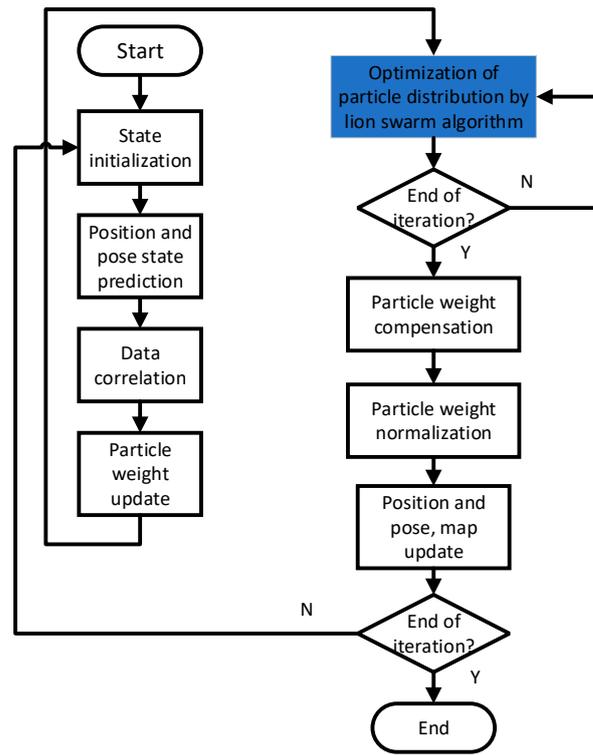


Figure 2. Flowchart of improved algorithm.

5. Performance Analysis

This paper first verifies the feasibility of the algorithm through the MATLAB simulation environment, and then verifies the feasibility of the LSOFastSLAM2.0 algorithm proposed in this paper under the simulated mine environment.

Simulation

The hardware environment of the experiment is a desktop computer (IntelCorei5 processor, 4 GB memory), and the experiment environment is MATLAB2016b.

First, a mobile robot simulation model is established, in which the mobile robot motion model can be represented as:

$$\begin{bmatrix} x_k^v \\ y_k^v \\ \Phi_k^v \end{bmatrix} = \begin{bmatrix} x_{k-1}^v + \Delta t \cdot \cos(\Phi_{k-1}^v + \theta_k) \\ y_{k-1}^v + \Delta t \cdot \sin(\Phi_{k-1}^v + \theta_k) \\ \Phi_{k-1}^v + \frac{\Delta t \cdot v_k \sin(\theta_k)}{D} \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \\ v_\Phi \end{bmatrix} \quad (14)$$

In the formula, (x_k^v, y_k^v) represents the position posture state of the robot in the two-dimensional environment at the k moment; Φ_k^v represents the heading angle, and the value range is $[-180^\circ, 180^\circ]$; v_k represents the movement speed of the robot, α_k represents its steering angle, and θ_k is the robot odometer sampling time, v_x, v_y, v_Φ is the noise during the robot movement, D is the distance between the drive shafts.

Then the observation model of the robot is established, and its formula is as follows:

$$\begin{bmatrix} r_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} \sqrt{(x_i - x_k^v)^2 + (y_i - y_k^v)^2} \\ \arctan \frac{y_i - y_k^v}{x_i - x_k^v} - \Phi_k^v \end{bmatrix} + \omega_k \quad (15)$$

In the formula, r_k, θ_k respectively represents the distance between the detected environmental feature and the mobile robot, and the angle of movement direction; ω_k is the observation noise.

The motion parameters and noise parameters of the mobile robot are shown in Table 1 below. The control noise of the robot and the observation noise are three-dimensional. In the following table, the speed noise of the robot represents the noise in the direction of the robot's movement process, which is composed of x -axis noise and y -axis noise.

Table 1. Motion parameters and noise parameters of mobile robots.

Parameter	Numerical Value	Noise Parameters	Numerical Value
Robot speed	3 m/s	Motion noise	0.3 m/s
Max steering angle	10°		1.5°
Maxi steering angular speed	15°/s	Observation noise	0.1 m/s
Wheel spacing	4 m		1°
Sampling time interval	0.025 s		

Secondly, simulate the robot localization and mapping, establish the working environment, as shown in Figure 3, set the 17 heading points, 35 road marking points and the mobile robot movement range is 100 m \times 80 m. The mobile robot starts from the origin of the coordinates (red point in Figure 3) and moves counterclockwise; the green * represents the road marking point, red \circ represents the heading point, and cyan line represents the specified robot path.

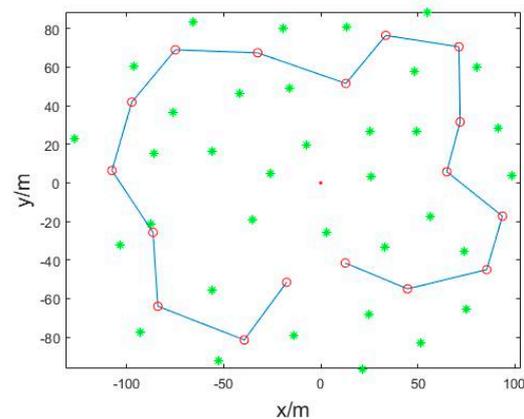


Figure 3. Simulation Environment.

Finally, in order to verify the effectiveness of the algorithm proposed in this paper, comparative algorithms were also performance which includes: the classic FastSLAM2.0 algorithm, the improved FastSLAM2.0 algorithm based on the gravitational field algorithm (GFA-FastSLAM2.0) [33], and the algorithm proposed in this paper (LSO-FastSLAM2.0). GFA-FastSLAM2.0 has been described in detail in reference [33], but for the sake of completeness, we will introduce its main idea. This algorithm introduces the optimization idea of gravitational field in particle resampling, and the sampled particles are regarded as the cosmic dust system. Each dust receives the action of the movement factor and the rotation factor of the central dust with the largest weight, so as to optimize the distribution of the sampling particles of the mobile robot pose so that the particle set can move towards the real particle more quickly. The pose state of the robot can be approximated to make it converge faster, and the particle degradation and depletion problems that are prone to occur in the FastSLAM2.0 algorithm are improved.

Since the motion noise and observation noise are random, in order to verify the feasibility of the improved algorithm proposed, this paper uses the root mean square error (RMSE) as the criterion. When the number of particles is 20, 50, 80, 100, The FastSLAM2.0 algorithm, GFA-FastSLAM2.0 algorithm, and LSO-FastSLAM2.0 algorithm were performed

20 experiments to obtain the RMSE of the robot position estimation and the road sign estimation, and further, take the average value. The experimental results are shown in Table 2 below.

Table 2. Improved algorithm validity proof.

Number of Particles	Algorithm	Mean Localization Accuracy Error/M	RMSE of Road Sign Estimation (m)
20	FastSLAM2.0	3.0535	4.1399
	GFA-FastSLAM2.0	2.9060	3.3545
	LSO-FastSLAM2.0	2.3025	2.2837
50	FastSLAM2.0	2.7718	3.2106
	GFA-FastSLAM2.0	2.3629	2.5990
	LSO-FastSLAM2.0	2.0470	2.2837
80	FastSLAM2.0	2.6843	2.9199
	GFA-FastSLAM2.0	1.7504	1.9072
	LSO-FastSLAM2.0	1.2745	1.3762
100	FastSLAM2.0	2.5907	2.8538
	GFA-FastSLAM2.0	1.3693	1.6422
	LSO-FastSLAM2.0	1.1745	1.3279

It can be seen from the above table, the improved algorithm proposed in this paper gradually decreases with the increase of the number of particles, and the positioning accuracy of the algorithm and the accuracy of the road signs gradually decrease. And with the increase of the number of particles, the filtering accuracy of the algorithm tends to stabilize, which is consistent with the convergence characteristics of the FastSLAM algorithm, proving the feasibility of an improved algorithm.

When the number of particles is 20, we verify the improvement degree of the algorithm proposed in this paper about the problem of particle degradation and compares the effective number of particles Neff in the three algorithms. As shown in Figure 4, the effective particle number of the algorithm proposed in this paper is higher than that of the other two algorithms. After calculation, in the process of positioning the three algorithms, the average number of effective particles is 13.8256, 16.3859, and 17.1674, respectively. This phenomenon may owe to the fact that the algorithm proposed in this paper effectively solves the problem of particle weight degradation, and GFA-FastSLAM2.0 algorithm. This phenomenon shows that the algorithm proposed in this paper efficiently optimizes the particle set and solve the problem of particle weight degradation.

In Figure 5, the green and red triangles represent the real and predicted positions of the robot, and the yellow lines represent the observation of the robot on the road markings. It can be seen from Figure 5 that the improved algorithm proposed in this paper has the highest degree of coincidence with the real trajectory, followed by GFA-FastSLAM2.0 algorithm, and the FastSLAM2.0 algorithm has the worst effect. This indicates that the algorithm proposed in this paper has the highest positioning accuracy among several algorithms. The reason for this phenomenon is that the FastSLAM2.0 algorithm has serious particle degradation and loss of particle diversity in the later stage of the algorithm. The GFA-FastSLAM2.0 algorithm has a higher filtering accuracy, because it acts on the particles through the gravitational field to distribute the particles in the high-likelihood area, which effectively alleviates the problem of particle degradation, and therefore improves the filtering accuracy of the robot. The LSO-FastSLAM2.0 algorithm has the highest positioning accuracy, compared with the optimization of the gravity field algorithm, the Lion algorithm is more effective in particle optimization, so it can effectively solve the problem of particle weight degradation and improve the filtering accuracy of the robot.

In order to further verify the optimization effect of PCSA-FastSlam2.0 on robot positioning and mapping, the predicted and estimated Euclidean distances of the three algorithms were compared at 8000 sampling points. The formula is:

$$\rho = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (16)$$

In the formula, $(x_1, y_1), (x_2, y_2)$ represents the coordinates of predicted position and actual position.

The comparison of the robot localization accuracy errors of the three algorithms are shown in Figure 6.

It can be seen from Figure 6 that the algorithm proposed in this paper has the smallest error and is relatively stable, while the positioning accuracy error of the classic FastSLAM2.0 algorithm gradually increases as the positioning accuracy error of the algorithm increases as the running time increases. The reason for this problem is that in the later period of the iterative algorithm, the particles are severely degraded and the diversity of the particles is lost, resulting in lower robot positioning accuracy. The improvement ideas proposed in this article effectively solve this problem and increase the positioning accuracy of the robot. The positioning accuracy of the GFA-FastSLAM2.0 algorithm is lower than that of the improved algorithm proposed in this paper, but it is better than the classic FastSLAM2.0 algorithm. This may be because of the improved particle filter by the gravity field algorithm. The gravity field algorithm has a certain degree of optimization for the particles after importance sampling, and a certain extent to improve the diversity of the particles.

After comparing the average error and variance of the positioning accuracy of the three algorithms, it is obvious that the improved algorithm has improved the robot positioning effect, as shown in Table 3 below.

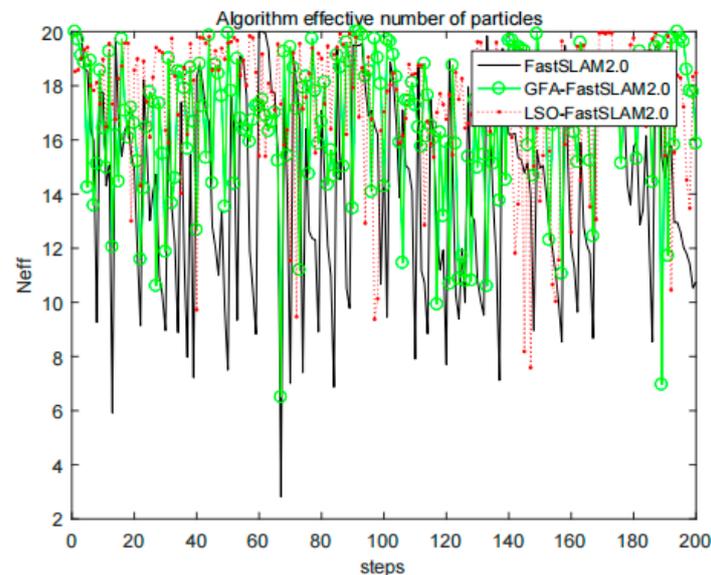


Figure 4. Effective particle number comparison.

Table 3. Comparison of mean error and variance of localization accuracy of three algorithms.

Algorithm	Mean Localization Accuracy Error/m	Variance of Localization Accuracy Error
FastSLAM2.0	2.7718	1.6403
GFA-FastSLAM2.0	1.4036	0.9059
LSO-FastSLAM2.0	1.1867	0.2519

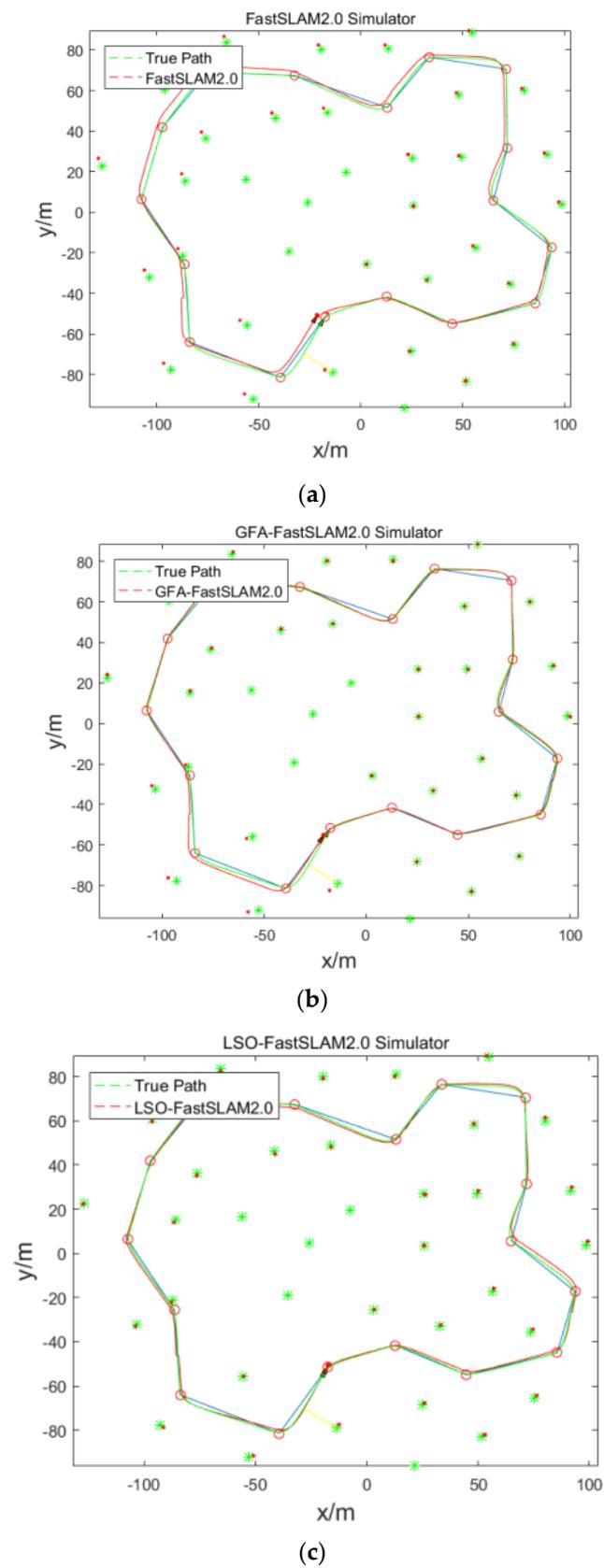


Figure 5. The algorithm real trajectory: (a) FastSLAM2.0 Simulation Result; (b) GFA-FastSLAM2.0 Simulation Result; (c) LSO-FastSLAM2.0 Simulation Result.

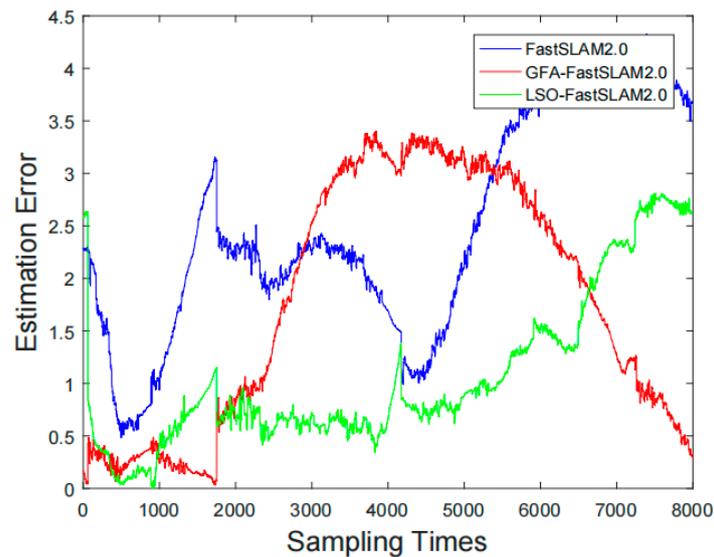


Figure 6. Comparison Chart of Robot Localization Accuracy Error.

In the above table, the LSO-FastSLAM2.0 algorithm has the smallest average error and the smallest positioning accuracy variance, indicating that the positioning accuracy and stability of the LSO-FastSLAM2.0 algorithm have been significantly improved. In the GFA-FastSLAM 2.0 algorithm, the strategy for sampling particles is adjusted so that the centers of the sampled particles are distributed near the global optimal value. Due to the unique strategy of simulating gravitational fields, the central dust attracts and repels the surrounding dust and adjusts the distribution of particles. Compared with the FastSLAM algorithm proposed in this paper, GFA-FastSLAM only attracts and repels particles, while LSO-FastSLAM divides particles into three different sets of the Lion King, Lioness, and young lion, and then through different formulas, the optimization of particle sets is realized, and the optimization process is more refined, so the accuracy and stability of the algorithm are improved.

In order to verify the degree of improvement in the accuracy of robot localization and mapping, we compared the RMSE of the x -axis, y -axis, and road signs, respectively, as shown in Table 4.

Table 4. Comparison of RMSE of three algorithms: x -axis, y -axis and road sign estimation.

Algorithm	RMSE of x -Axis (m)	RMSE of y -Axis (m)	RMSE of Road Sign Estimation (m)
FastSLAM2.0	2.0447	2.2676	2.9871
GFA-FastSLAM2.0	1.6015	1.1018	1.5841
LSO-FastSLAM2.0	0.6932	1.0518	1.3383

It can be seen from Table 4 that the improved algorithm proposed in this paper is better than FastSLAM2.0 and GFA-FastSLAM2.0 algorithms in the x -axis, y -axis, and road sign estimation. Compared to the FastSLAM2.0 algorithm, the LSO-FastSLAM2.0 algorithm significantly improves the robot positioning and mapping. Compared with GFA-FastSLAM2.0, LSO-FastSLAM2.0 also has a significant improvement, which shows that the algorithm proposed in this paper more effectively optimizes the importance sampling process so as to achieve a significant improvement of the robot positioning map building accuracy.

6. Conclusions

In order to improve the accuracy of mine rescue robot positioning and mapping, this paper proposes an improved FastSLAM algorithm based on the lion swarm optimization algorithm. The lion swarm optimization algorithm optimizes the particle distribution after sampling so that the particle set is distributed in the high-likelihood area, thus solving the particle weight degradation and loss of particle diversity in the FastSLAM algorithm thereby improving the accuracy of robot positioning and mapping. The robot simultaneous positioning and mapping experiments show that the improved algorithm not only improves the positioning accuracy of the robot but also improves the algorithm stability. In future work, the improved algorithm will be applied to mine rescue robots to further verify the feasibility of the algorithm.

Author Contributions: Conceptualization, D.Z.; Data curation, M.W. and Y.M.; Formal analysis, Y.M., J.Y., S.L. and Y.Y.; Investigation, M.W.; Methodology, Y.M. and J.Y.; Project administration, D.Z. and Y.M.; Software, Y.M. and M.W.; Supervision, D.Z.; Writing—original draft, D.Z. and Y.M.; Writing—review and editing, J.Y. and Y.M. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the National Natural Science Foundation of China (51774235) and the Shaanxi Provincial Key R&D General Industrial Project (2021GY-338).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the National Natural Science Foundation of China and the Shaanxi Provincial Key R&D General Industrial Project.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Hua, Z.; Shaoze, Y. Research and experiment of a new type of coal mine rescue robot. *J. China Coal Soc.* **2020**, *45*, 2170–2181.
2. Qian, S.-H.; Ge, S.-R.; Wang, Y.-S.; Wang, Y.; Liu, C.-Q. Research status of the disaster rescue robot and its applications to the mine rescue. *Robot* **2006**, *28*, 350–354.
3. Ge, S.; Hu, E.; Pei, W. Classification system and key technology of coal mine robot. *J. China Coal Soc.* **2020**, *45*, 455–463.
4. Durrant, W.; Hugh, F.; Bailey, T. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robot. Autom. Mag.* **2006**, *13*, 99–110. [[CrossRef](#)]
5. Guivant, J.E.; Nebot, E.M. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Trans. Robot. Autom.* **2002**, *17*, 242–257. [[CrossRef](#)]
6. Agarwal, S.; Shree, V.; Chakravorty, S. RFM-SLAM: Exploiting relative feature measurements to separate orientation and position estimation in SLAM. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017.
7. Chen, H.; Yu, P. Application of Laser SLAM Technology in Backpack Indoor Mobile Measurement System. *IOP Conf. Ser. Earth Environ. Sci.* **2019**, *237*, 032026. [[CrossRef](#)]
8. Nemra, A.; Aouf, N. Robust cooperative UAV Visual SLAM. In Proceedings of the IEEE International Conference on Cybernetic Intelligent Systems, Hangzhou, China, 26–27 August 2011.
9. Lina, M.; Tardos, J.D.; Neira, J. Divide and Conquer: EKF SLAM in $O(n)$. *IEEE Trans. Robot.* **2008**, *24*, 1107–1120.
10. Montemerlo, M.; Thrun, S.; Koller, D.; Wegbreit, B. FastSLAM: A Factored Solution to Simultaneous Mapping and Localization. In Proceedings of the National Conference on Artificial Intelligence, Bellevue, WA, USA, 14–18 July 2013.
11. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J.J. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Trans. Robot.* **2016**, *32*, 1309–1332. [[CrossRef](#)]
12. Havangi, R. Improved FastSLAM2.0 using ANFIS and PSO. *Automatika* **2016**, *57*, 996–1006. [[CrossRef](#)]
13. Zhang, Y.; Zheng, X.F.; Luo, Y. SLAM algorithm with Gaussian distributed resampling Rao-Blackwellized particle filter. *Control Decis.* **2016**, *31*, 2299–2304.
14. Roy, S.S.; Pratihari, D.K. Dynamic modeling, stability and energy consumption analysis of a realistic six-legged walking robot. *Robot. Comput. Integr. Manuf.* **2013**, *29*, 400–416. [[CrossRef](#)]

15. Lv, T.; Feng, M. A FastSLAM 2.0 algorithm based on FC&ASD-PSO. *Robotica* **2016**, *35*, 1–21.
16. Ankişhan, H.; Ari, F.; Tartan, E.Ö.; Pafkfiliz, A.G. Square root central difference-based FastSLAM approach improved by differential evolution. *Turk. J. Electr. Eng. Comput. Sci.* **2016**, *24*, 994–1013. [[CrossRef](#)]
17. Kim, C.; Sakthivel, R.; Chung, W.K. Unscented FastSLAM: A Robust and Efficient Solution to the SLAM Problem. *IEEE Trans. Robot.* **2008**, *24*, 808–820. [[CrossRef](#)]
18. He, B.; Ying, L.; Shang, S.; Feng, X.; Yan, T.; Nian, R.; Shen, Y. Autonomous navigation based on unscented-FastSLAM using particle swarm optimization for autonomous underwater vehicles. *Measurement* **2015**, *71*, 89–101. [[CrossRef](#)]
19. Kurt-Yavuz, Z.; Yavuz, S. A comparison of EKF, UKF, FastSLAM2.0, and UKF-based FastSLAM Algorithms. In Proceedings of the IEEE International Conference on Intelligent Engineering Systems, Lisbon, Portugal, 13–15 June 2012.
20. Darwish, S.M. Combining firefly algorithm and Bayesian classifier: New direction for automatic multilabel image annotation. *IET Image Process.* **2016**, *10*, 763–772. [[CrossRef](#)]
21. Do, K.D. Stability in probability and inverse optimal control of evolution systems driven by Levy processes. *IEEE/CAA J. Autom. Sin.* **2020**, *7*, 405–419. [[CrossRef](#)]
22. Anwar, N.; Deng, H. Ant Colony Optimization based multicast routing algorithm for mobile ad hoc networks. In Proceedings of the Advances in Wireless & Optical Communications, Riga, Latvia, 5–6 November 2015.
23. Kharwar, P.K.; Verma, R.K. Exploration of nature inspired grey wolf algorithm and grey theory in machining of multiwall carbon nanotube/polymer nanocomposites. *Eng. Comput.* **2020**, 1–22. [[CrossRef](#)]
24. Sengupta, S.; Das, S.; Nasir, M.; Vasilakos, A.V.; Pedrycz, W. Energy-Efficient Differentiated Coverage of Dynamic Objects Using an Improved Evolutionary Multi-Objective Optimization Algorithm with Fuzzy-Dominance. In Proceedings of the 2012 IEEE Congress on Evolutionary Computation, Brisbane, Australia, 10–15 June 2012.
25. Lee, H.-C.; Park, S.-K.; Choi, J.-S.; Lee, B.-H. PSO-FastSLAM: An Improved FastSLAM Framework Using Particle Swarm Optimization. In Proceedings of the 2009 IEEE International Conference on Systems, Man and Cybernetics, San Antonio, TX, USA, 11–14 October 2009.
26. Haydar, A.; Tartan, E.Ö.; Ari, F. Square Root Unscented Based FastSlam Optimized by Particle Swarm Optimization Passive Congregation. In Proceedings of the IEEE International Conference on Mechatronics and Automation, Takamatsu, Japan, 4–7 August 2013.
27. Lv, T.-Z.; Zhao, C.-X.; Zhang, H.-F. An Improved FastSLAM Algorithm Based on Revised Genetic Resampling and SR-UPF. *Int. J. Autom. Comput.* **2018**, *15*, 325–334. [[CrossRef](#)]
28. Cui, J.; Feng, D.; Li, Y.; Tian, Q. Research on simultaneous localization and mapping for AUV by an improved method: Variance reduction FastSLAM with simulated annealing. *Def. Technol.* **2019**, *16*, 651–661. [[CrossRef](#)]
29. Tian, M.-C.; Bo, Y.-M.; Chen, Z.-M.; Yue, C.; Wang, H. Novel target tracking method based on firefly algorithm optimized particle filter. *Control Decis.* **2017**, *32*, 1758–1766.
30. Zhang, Q.-M.; Liu, L.-Q.; Ma, L.-Q. New Swarm Intelligent Algorithms: Lions Algorithm. *Comput. Sci.* **2018**, *45*, 114–116.
31. Montemerlo, M.S.; Thrun, S.B.; Roller, D.; Wegbreit, B. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. *Proc. Int. Conf. Artif. Intell.* **2003**, *133*, 1151–1156.
32. Yang, X.; Jiao, Q.; Liu, X. Research on GRNN-based Particle Filter Algorithm. In Proceedings of the Information Technology, Networking, Electronic and Automation Control Conference, Chengdu, China, 15–17 March 2019.
33. Chen, S.-M.; Liu, J.-K.; Xiao, J. Unscented Fast SLAM2.0 algorithm based on gravitational field optimization. *Control Theory Appl.* **2018**, *35*, 1186–1193.